pi_circle_estimation

March 31, 2021

1 Estimation of Pi value with random points

@date: 14/03/2021

@Source: based on @picanumeros

@author: Javier Cebrián Casado

[2]:

```
import numpy as np
import plotly.graph_objects as go
import plotly.io as pio
pio.renderers.default = "notebook+pdf"
```

1.1 Calculation of n random numbers in a x[-2,2] y[-2,2] plane.

Pi value is aproximate with the number of points in a r=2 circle:

$$\pi = \frac{r^2}{area} \approx \frac{innerpoints}{totalpoints} r^2$$

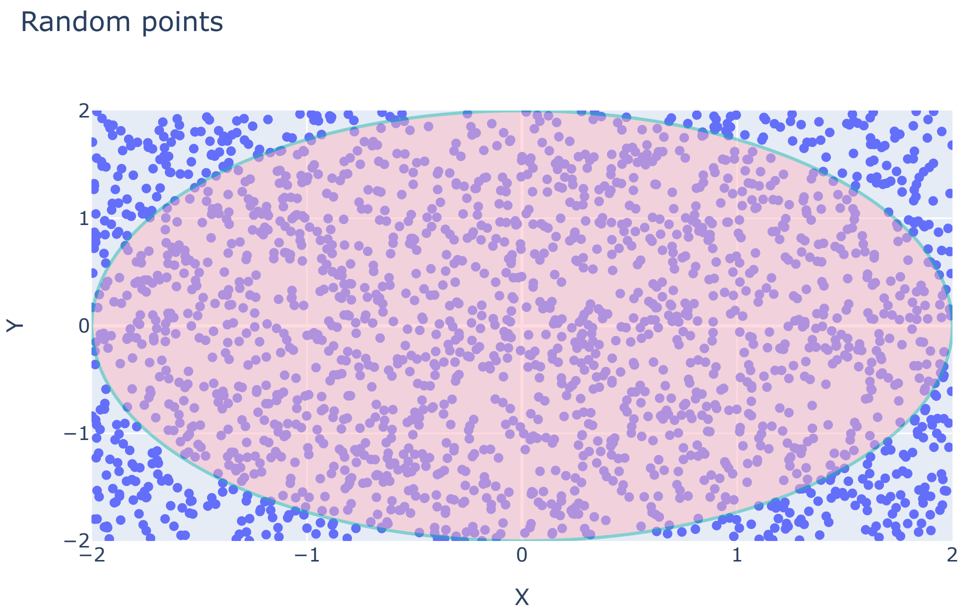
[3]:

```
numberPoints=2000
points=np.random.rand(2,numberPoints)*4-2
innerPoints=np.sum((points[0,:]**2+points[1,:]**2)<=4)
print('Estimated pi is: '+str((innerPoints/numberPoints)*4))

data1 = go.Scatter( x=points[0,:],y=points[1,:],mode='markers',name='Points')

layout = go.Layout(title='Random points')
fig = go.Figure(data= data1, layout = layout)
fig.update_xaxes(range=(-2,2), title='X')
fig.update_yaxes(range=(-2,2),title='Y')
fig.add_shape(type="circle",
    xref="x", yref="y",
    fillcolor="lightpink",
    x0=-2, y0=-2, x1=2, y1=2,
    line_color="LightSeaGreen",
    opacity=0.5,
    #layer="below"
)
fig.show()
```

```
Estimated pi is: 3.062
```


Random points

1.2 Error of estimation

Error in function of the number of points.

$$Error = \mid \pi - \hat{\pi} \mid$$

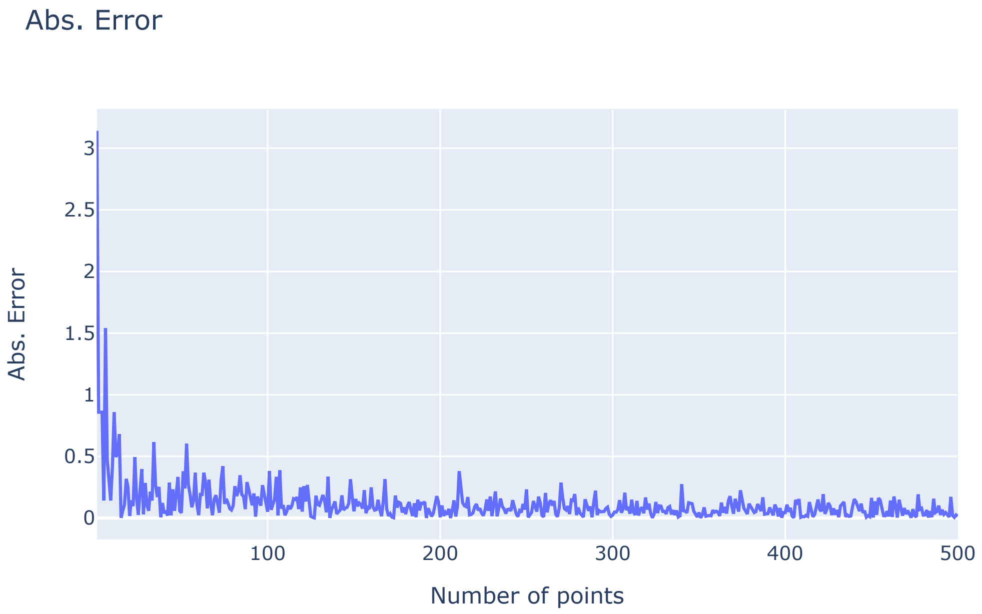
[4]:

```
pi=np.array(0)
numberPoints=500
for i,j in enumerate(range(1,numberPoints)):
    points=np.random.rand(2,j)*4-2
    innerPoints=np.sum((points[0,:]**2+points[1,:]**2)<=4)
    pi=np.append(pi,(innerPoints/j)*4)

error=np.abs(np.pi-pi)

data1 = go.Scatter( x=np.
 ↪linspace(1,numberPoints,numberPoints),y=error,mode='lines',name='Abs. Error')
data = [data1]
layout = go.Layout(title='Abs. Error')
```

```
fig = go.Figure(data= data, layout = layout)
fig.update_xaxes(title='Number of points')
fig.update_yaxes(title='Abs. Error')
fig.show()
```



Percent error:

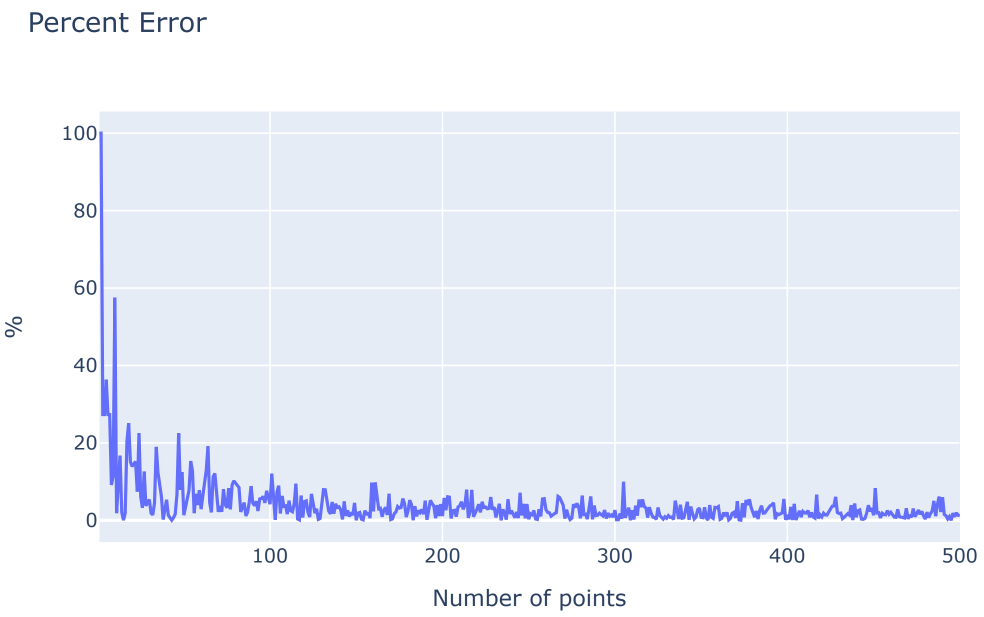
$$\%Error = \frac{\mid \hat{\pi} - \pi \mid}{\pi}$$

[7]:
```
pi=np.array(0)
numberPoints=500
for i,j in enumerate(range(1,numberPoints)):
    points=np.random.rand(2,j)*4-2
    innerPoints=np.sum((points[0,:]**2+points[1,:]**2)<=4)
    pi=np.append(pi,(innerPoints/j)*4)

error=(np.abs(np.pi-pi)/np.pi)*100

data1 = go.Scatter( x=np.
 ↪linspace(1,numberPoints,numberPoints),y=error,mode='lines',name='Percent␣
 ↪Error')
data = [data1]
layout = go.Layout(title='Percent Error')
fig = go.Figure(data= data, layout = layout)
```

```
fig.update_xaxes(title='Number of points')
fig.update_yaxes(title='%')
fig.show()
```



```
[ ]:
```